Data Fusion in Tinyml and Applications in Biology and Federated Learning

Claudio Miceli de Farias - Universidade Federal do Rio de Janeiro
cmicelifarias@cos.ufrj.br
Internet Of Things
Industry 5.0

<table>
<thead>
<tr>
<th></th>
<th>1st</th>
<th>2nd</th>
<th>3rd</th>
<th>4th</th>
<th>5th</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mechanization, water power, steam power</td>
<td>Mass production, assembly line, electricity</td>
<td>Computer and automation</td>
<td>Cyber Physical Systems</td>
<td>Mass customization &amp; cyber physical cognitive systems</td>
</tr>
</tbody>
</table>

5.0 vs 4.0
TinyML - Computer Vision

- Tensor flow lite micro
TensorFlow Lite

TinyML

[1] Training
[2] Distillation
[3] Quantization
[4] Encoding
[5] Compilation
Problem - not a new one

- Resource constrained environment
- Decision making
- CNN is the traditional way
Weightless Neural Networks

- A different type of Neural Network
- Low cost
- Without weights :)
- Wisard architecture - bethoween has been proposed as a related work!

Figure 1. A WiSARD discriminator.
Proposal

Apply WNN into Computer Vision for Edge AI and compare with traditional techniques!
Simulation

- 2 setups
  - Google colab - baseline
  - Different boards
    - NVIDIA Jetson Nano 2GB
    - Google Coral Dev Board
    - Raspberry Pi 4Gb + Intel Neural Compute Stick 2
    - Raspberry Pi 4Gb + Google Coral TPU
    - Raspberry Pi 4Gb
Dataset 1 - Human Activity Recognition
## Model Summary

### Model: "sequential"

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output Shape</th>
<th>Param #</th>
</tr>
</thead>
<tbody>
<tr>
<td>vgg16 (Functional)</td>
<td>(None, 512)</td>
<td>14714688</td>
</tr>
<tr>
<td>flatten (Flatten)</td>
<td>(None, 512)</td>
<td>0</td>
</tr>
<tr>
<td>dense (Dense)</td>
<td>(None, 512)</td>
<td>262656</td>
</tr>
<tr>
<td>dense_1 (Dense)</td>
<td>(None, 15)</td>
<td>7695</td>
</tr>
</tbody>
</table>

Total params: 14,985,039
Trainable params: 270,351
Non-trainable params: 14,714,688
Results - Loss and Accuracy
JETSON NANO DEVELOPER KIT
### TABLE I
**Classification Report - Dataset 1 - Input size: 64 x 64**

<table>
<thead>
<tr>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>calling</td>
<td>0.2143</td>
<td>0.1364</td>
<td>0.1667</td>
<td>154</td>
</tr>
<tr>
<td>clapping</td>
<td>0.4113</td>
<td>0.3473</td>
<td>0.3766</td>
<td>167</td>
</tr>
<tr>
<td>cycling</td>
<td>0.5107</td>
<td>0.7346</td>
<td>0.6025</td>
<td>162</td>
</tr>
<tr>
<td>dancing</td>
<td>0.3438</td>
<td>0.2865</td>
<td>0.3125</td>
<td>192</td>
</tr>
<tr>
<td>drinking</td>
<td>0.2542</td>
<td>0.1875</td>
<td>0.2158</td>
<td>160</td>
</tr>
<tr>
<td>eating</td>
<td>0.3209</td>
<td>0.6013</td>
<td>0.4185</td>
<td>158</td>
</tr>
<tr>
<td>fighting</td>
<td>0.4500</td>
<td>0.3387</td>
<td>0.3865</td>
<td>186</td>
</tr>
<tr>
<td>hugging</td>
<td>0.2432</td>
<td>0.2711</td>
<td>0.2564</td>
<td>166</td>
</tr>
<tr>
<td>laughing</td>
<td>0.3529</td>
<td>0.3313</td>
<td>0.3418</td>
<td>163</td>
</tr>
<tr>
<td>listening_to_music</td>
<td>0.2323</td>
<td>0.2130</td>
<td>0.2222</td>
<td>169</td>
</tr>
<tr>
<td>running</td>
<td>0.3839</td>
<td>0.4355</td>
<td>0.4081</td>
<td>186</td>
</tr>
<tr>
<td>sitting</td>
<td>0.2153</td>
<td>0.2500</td>
<td>0.2314</td>
<td>180</td>
</tr>
<tr>
<td>sleeping</td>
<td>0.5533</td>
<td>0.5000</td>
<td>0.5253</td>
<td>166</td>
</tr>
<tr>
<td>texting</td>
<td>0.1923</td>
<td>0.1212</td>
<td>0.1487</td>
<td>165</td>
</tr>
<tr>
<td>using_laptop</td>
<td>0.2723</td>
<td>0.3059</td>
<td>0.2881</td>
<td>170</td>
</tr>
<tr>
<td><strong>accuracy</strong></td>
<td></td>
<td></td>
<td>0.3369</td>
<td>2544</td>
</tr>
<tr>
<td><strong>macro avg</strong></td>
<td>0.3301</td>
<td>0.3373</td>
<td>0.3267</td>
<td>2544</td>
</tr>
<tr>
<td><strong>weighted avg</strong></td>
<td>0.3313</td>
<td>0.3369</td>
<td>0.3274</td>
<td>2544</td>
</tr>
</tbody>
</table>

### TABLE II
**Classification Report - Dataset 2**

<table>
<thead>
<tr>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>WALKING</td>
<td>0.89</td>
<td>0.85</td>
<td>0.87</td>
<td>259</td>
</tr>
<tr>
<td>UPSTAIRS</td>
<td>0.76</td>
<td>0.89</td>
<td>0.82</td>
<td>216</td>
</tr>
<tr>
<td>DOWNSTAIRS</td>
<td>0.92</td>
<td>0.81</td>
<td>0.87</td>
<td>210</td>
</tr>
<tr>
<td>SITTING</td>
<td>0.67</td>
<td>0.75</td>
<td>0.71</td>
<td>261</td>
</tr>
<tr>
<td>STANDING</td>
<td>0.73</td>
<td>0.65</td>
<td>0.69</td>
<td>270</td>
</tr>
<tr>
<td>LAYING</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>256</td>
</tr>
<tr>
<td><strong>accuracy</strong></td>
<td></td>
<td></td>
<td>0.82</td>
<td>1472</td>
</tr>
<tr>
<td><strong>macro avg</strong></td>
<td>0.83</td>
<td>0.82</td>
<td>0.82</td>
<td>1472</td>
</tr>
<tr>
<td><strong>weighted avg</strong></td>
<td>0.83</td>
<td>0.82</td>
<td>0.82</td>
<td>1472</td>
</tr>
</tbody>
</table>
Conclusions and Future works

- WNN are effective for Edge AI
- The results had close results to traditional CNN
- As future works:
  - test different types of neural network as classifiers for human action recognition;
  - develop a technique that uses a WNN as input to adapt to network conditions;
  - use a WNN to try to fix distortions in a video in realtime;
LUA and the RoboSub Competition
LUA’s Software Design

**Sense**
- RTAB-MAP
- Darknet
- Acoustic Beamform
- Kalman Filter

**Plan**
- Finite State Machine
- Path Planning

**Act**
- PID Control
- Thrusters and Actuators

**ROS Middleware**

**Linux**
Proposal

- A fast implementation of the beamforming algorithm in the time and frequency domains
- Errors:
  - (i) the first derives from the noise of the sensors and signals,
  - (ii) arrangement of the sensors, which is a function of the true azimuth and elevation angles.

- Able to learn about angles combination
- Two approaches:
  - (i) a Convolutional Neural Network and
  - (ii) a clustering algorithm.
/* Training */
b = Beamforming()
X = List()
y = List()
for sound in sounds do
  RMS = b.frequency_beamforming(sound)
  X.append(RMS)
  (az_pred, el_pred) = b.angle_of_arrival(sound)
  error = (az_real - az_pred, el_real - el_pred)
  y.append(error)
end for
model = NeuralNetwork()
model.train(X, y)
/* Usage */
sound = new_sound()
RMS = b.frequency_beamforming(sound)
error_pred = model.predict(RMS)
(az, el) = b.angle_of_arrival(sound) - error_pred
/* Training */
1: b = Beamforming()
2: angles = [(az, el) for az in range(361) for el in range(181)]
3: kmeans = KMeans(number_of_clusters).fit(angles)
4: errors = HashMap(k: List() for k in range(number_of_clusters))
5: for sound in sounds do
6:   (az_pred, el_pred) = b.angle_of_arrival(sound)
7:   angles_cluster = kmeans.predict(((az_pred, el_pred))
8:   error = (az_real - az_pred, el_real - el_pred)
9:   errors[angles_cluster].append(error)
10: end for
11: median_errors = HashMap(k: median(errs) for k, errs in errors)
12: /* Usage */
13: sound = new_sound()
14: (az_pred, el_pred) = b.angle_of_arrival(sound)
15: angles_cluster = kmeans.predict(((az_pred, el_pred))
16: error_pred = median_errors[angles_cluster]
17: (az, el) = (az_pred, el_pred) - error_pred
Experimental Design

- Nvidia Jetson Nano
- Python
- ROS
- 30 repetitions
- 95% confidence interval
- Synthetic and real data
Experimental Design

SLAM

Path Planning
## Experimental Design

<table>
<thead>
<tr>
<th>Beamforming</th>
<th>Neural Network</th>
<th>Clusterization</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.88</td>
<td>12.18</td>
<td>9.36</td>
</tr>
<tr>
<td>0.00%</td>
<td>-23.28%</td>
<td>5.26%</td>
</tr>
</tbody>
</table>

**TABLE I**
SELD [17] DATASET EVALUATION MEAN ABSOLUTE ERROR - AZIMUTH

<table>
<thead>
<tr>
<th>Beamforming</th>
<th>Neural Network</th>
<th>Clusterization</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.45</td>
<td>5.57 ± 0.38</td>
<td>13.18 ± 0.13</td>
</tr>
<tr>
<td>0.00%</td>
<td>66.14%</td>
<td>19.89%</td>
</tr>
</tbody>
</table>

**TABLE II**
SYNTHETIC DATASET EVALUATION MEAN ABSOLUTE ERROR - AZIMUTH
Conclusions

- Lua, a low-cost AUV developed by the UFRJ Nautilus team
- Software components of the AUV
- New beamforming algorithms
- Use time and frequency simultaneously
Future works

- Increase the amount of training data and/or different Network architectures.
- A better synthetic dataset generator
- Correlation between $\mu$ and statistical metrics
- Federated Learning among a group of AUV
Internet of Bionanomachines
Long range molecular communication

Wireless Options
- Pheromones
  - Pollen / Spores
- Light transduction

Wired Options
- Axons
- Capillaries
Traditional communication

Communication carrier:
   Electromagnetic wave
Signal type:
   Electronic and optical signal
Propagation speed:
   Light speed \((3 \times 10^5 \text{Km/s})\)
Propagation environment:
   Airborne medium
Encoded information:
   Voice, text, and video
Behavior of receiver:
   A receiver interprets encoded information
Other features:
   Accurate communication and high energy consumption

Molecular communication

Communication carrier:
   Molecule
Signal type:
   Chemical signal
Propagation speed:
   Extremely slow speed
Propagation environment:
   Aqueous medium
Encoded information:
   Phenomena and chemical states
Behavior of receiver:
   Information molecules cause chemical reactions at a receiver
Other features:
   Stochastic communication and low energy consumption

Different communication paradigms
So, we are proposing a low-cost interface for long-range IoBNT communication through indirect sending using High-Level Data Fusio
Architecture Design

Conversion Unit

External Comm

Detection Threshold

Biochemical Signal

Interface Manager

Inference Parameter

IP

External Comm

CU

IM

EC
External entity populates Inference Parameter Database

IM senses the environment

IM Applies a filter

Sends result to the EC

EC sends to an external system

CU applies a BN

CU consults the IP
Bayesian Networks

\[ P(X) = \prod_{i=1}^{n} P(X_i | \text{parents}(X_i)). \] (1)
Bayesian Network for Ethylene Concentration Prediction

Temperature

Is_Raining

Humidity

Observed_Concentration
Conclusions and Future Works

- Framework based on indirect sensing that uses high level information fusion techniques to build gateways to Internet of Bionano Things.
- Use indirect sensing measurements and high level information fusion techniques to infer about communications status.

As future works we intend to:

1. Explore new DFTs to improve accuracy;
2. Use Machine Learning techniques to predict the environment behavior;
3. Use the indirect sensing to enhance the biological sensor decisions by combining different types of measurements.
local updates \rightarrow \text{new global model} \rightarrow \text{local updates}

local data

Subject: Thank you for the feedback
Data Fusion in Tinyml and Applications in Biology and Federated Learning

Claudio Miceli de Farias - Universidade Federal do Rio de Janeiro
cmicelifarias@cos.ufrj.br